

객체 추적을 위한 보틀넥 기반 Siam-CNN 알고리즘

임수창[†], 김종찬^{††}

Bottleneck-based Siam-CNN Algorithm for Object Tracking

Su-Chang Lim[†], Jong-Chan Kim^{††}

ABSTRACT

Visual Object Tracking is known as the most fundamental problem in the field of computer vision. Object tracking localize the region of target object with bounding box in the video. In this paper, a custom CNN is created to extract object feature that has strong and various information. This network was constructed as a Siamese network for use as a feature extractor. The input images are passed convolution block composed of a bottleneck layers, and features are emphasized. The feature map of the target object and the search area, extracted from the Siamese network, was input as a local proposal network. Estimate the object area using the feature map. The performance of the tracking algorithm was evaluated using the OTB2013 dataset. Success Plot and Precision Plot were used as evaluation matrix. As a result of the experiment, 0.611 in Success Plot and 0.831 in Precision Plot were achieved.

Key words: Deep Learning, Computer Vision, Object Tracking, Convolutional Neural Networks, Siamese Network

1. 서 론

객체 추적(object tracking)은 컴퓨터 비전 분야에서 중요한 연구 분야이다. 객체 추적은 비디오 감시, 자율주행 자동차, 드론과 같이 카메라를 사용하는 시각 응용 어플리케이션 모든 분야의 핵심 기술이다 [1-3]. 객체 추적은 일반적으로 비디오 입력, 객체 영역 초기화(Initialization), 특징을 사용한 외형 모델(Appearance Model) 생성, 영역 추정, 영역 지역화의 과정으로 진행된다. 객체 영역 초기화는 연속된 비디오 프레임의 첫 번째 프레임에서 객체 영역을 추출하는 과정이다. 추출된 객체 영역은 수학적 모델링 또는 색상, 윤곽선 추출 등의 알고리즘을 사용하여 특징을 표현하고 이를 외형 모델로 생성한다. 외형 모

델을 사용하여 후속 프레임에서 객체의 위치를 찾고 추적한다. 객체 추적은 처음 입력된 객체의 영역만을 사용하여 후속 영상 시퀀스에서 객체를 추적한다. 객체의 형태 및 비디오 환경은 매 프레임마다 변화한다. 이 변화는 장애물에 의한 객체 가려짐, 객체 크기 변화(Scale variation), 주변 배경 또는 목표 객체 영역의 밝기 변화, 노이즈 또는 열화로 인한 복잡한 배경(background clutters) 등의 여러 환경 변화 요소를 포함한다. 이 요소들로 인해 비디오의 객체가 항상 초기 상태와 동일하다는 것을 보장받을 수 없으므로 객체 추적에 실패하는 표류(drift) 현상의 원인이 된다. 이런 문제들을 해결하기 위해 다양한 추적 알고리즘이 제안됐다. 이 알고리즘은 서로 상이한 이론적 근거에 기초하여 환경 변화에 영향을 최소화하므로

* Corresponding Author : Jong-Chan Kim, Address: (57922) Jungang-ro 255, Suncheon, Jeonnam, Korea, TEL : +82-61-750-3620, FAX : +82-61-750-3620, E-mail : seaghost@scnu.ac.kr
Receipt date : Dec. 6, 2021, Revision date : Jan. 3, 2022
Approval date : Jan. 4, 2022

[†] Dept. of Computer Engineering, Suncheon National University
(E-mail : suchanglim@suncheon.ac.kr)

^{††} Dept. of Computer Engineering, Suncheon National University

받으며 추적 정확도를 향상하는 것을 목표로 제작됐다[4].

기존 추적 알고리즘은 수작업(Hand-crafted) 방식의 특징 추출 알고리즘을 사용한다. 색상(Color), 질감(Texture), 모양(Shape) 및 윤곽선(Contour) 등의 다양한 특징으로 구성된다. 이러한 수작업 특징에 기반한 많은 추적 알고리즘이 제안되었다. 고유기저(Eigenbasis) 공간에서 압축 샘플링을 적용한 추적 알고리즘과 파티클 필터 등을 사용한 알고리즘이 제안되었다[5-6]. 수작업 특징 추출 알고리즘을 사용한 추적 알고리즘은 특정 영상 환경에 제한되어 있다. 대표적으로 특정 색상 특징을 사용한 알고리즘은 목표 객체가 균일하게 분포된 색상을 가져야 하는 조건을 지니고 있다[7]. 수작업 특징을 사용한 추적 알고리즘은 사전에 정의된 영상 환경에 영향을 주는 노이즈에 매우 민감하다. 노이즈는 잡음 또는 목표 객체와 유사한 배경 또는 객체를 포함한다. 따라서, 초기에 구축한 특징 정보에 따라 이미지에서 객체를 찾아내는 성능이 달라진다.

최근 연구들은 딥러닝(Deep learning)에 기반한 심층 특징(Deep feature)에 초점을 맞추고 있다. 심층 특징은 수작업 방법을 통해 추출된 특징에 비해 매우 많은 표현력을 지닌 정보로 구성되어 있다. 대표적으로, 심층 특징을 추출하는 CNN(Convolutional Neural Network)가 객체 추적 연구에 적용되었다[8]. CNN에서 추출되는 고수준 특징을 사용하여 객체 추적 알고리즘을 크게 개선할 수 있다. 목표 객체 이미지는 적층된 레이어를 거치며 저수준에서 고수준의 심층 특징으로 변환된다.

추적 문제를 유사성 비교 문제로 간주하는 접근 방식이 주목받고 있다. 목표 객체를 찾기 영역에 존재하는 객체를 유사성 비교를 통해 위치 및 영역을 추정하는 것이다. 대표적으로 삼 네트워크(Siamese network)를 사용하여 입력 프레임에서 대상 객체의 유사성을 비교하도록 학습한다. 이 방식은 종단간(End-to-End) 학습이 가능하다. 결과적으로 삼 네트워크 추적 방식 미세 조정 또는 온라인 업데이트 프로세스 없이 높은 정확도를 얻을 수 있다. 추적 알고리즘의 성능은 삼 네트워크에서 추출된 특징에 크게 의존한다.

본 논문에서는 객체 특징 강조 컨볼루션 블록 적용한 삼 네트워크 기반 추적 알고리즘을 제안한다.

특징 강조 및 추출을 위해 커스텀(Custom) CNN을 구현하였고, 객체 영역 추정을 위해 지역 제안 네트워크(Region Proposal Network)를 결합하였다. 제안하는 네트워크는 ILSVRC(ImageNet Large Scale Visual Recognition Challenge) 2015 데이터셋을 사용하여 학습했다. OTB-2013 데이터셋을 사용하여 정확도와 정밀도를 평가했다. 논문의 구성은 다음과 같다. 2장에서 딥러닝 기반 추적 알고리즘과 삼넷 기반 추적 알고리즘의 연구를 기술했다. 3장에서 제안한 알고리즘을 서술하며, 4장에서 성능평가를 진행하였다. 마지막 5장에서 결론을 맺는다.

2. 관련 연구

2.1 CNN기반 추적 알고리즘

객체 외형 표현(Appearance Representation)은 물체 추적에 매우 중요하다. 고전 추적 알고리즘은 윤곽선 및 색상 히스토그램과 같이 목표 객체 형태를 표현하기 위해 수작업 방식의 다양한 특징을 사용했다. 최근에는 딥러닝을 추적 알고리즘에 도입하는 연구가 진행되었다. 다 계층(Multi layer) 오토인코더(Auto Encoder)기반 네트워크를 사용하는 DLT(Deep Learning Tracker)가 대표적이다[9]. 최근 트렌드는 영상 분류, 물체 감지 등 다양한 컴퓨터 비전 작업에 CNN이 핵심 네트워크로 사용된다[10]. 다양한 비전 분야에서 보여준 CNN이 지닌 잠재적인 성능에 따라 객체 추적에 CNN이 가진 강력한 특징 표현(Representation)을 적용하려는 연구가 진행되고 있다. 이 특징은 원하는 대상을 찾기 위해 충분한 고수준 의미 정보로 구성된다. 분류문제를 위해 구현된 ResNet[11]과 같이 다른 비전 작업에 대해 사전 훈련된 네트워크의 특징을 사용한다. CNN을 사용함으로써 수작업 특징을 사용하는 고전적인 추적 알고리즘보다 향상된 성능을 보여주었다. CNN과 SVM(support vector machine)을 결합한 객체 추적 알고리즘은 목표 객체 외형 모델을 차별적으로 학습하기 위해 CNN을 특징 추출기로 사용하였다[12]. HDT(Hedged Deep Tracker) 추적 알고리즘 또한 CNN을 특징 추출기로 사용했다[13]. CNT(Convolutional Network based Tracker)는 적응적(adaptive) 필터를 사용하여 목표 객체 영역에서 특징 패치를 추출하여 객체 구조 정보를 인코딩하기 위한 보조 자료로 구축한다.

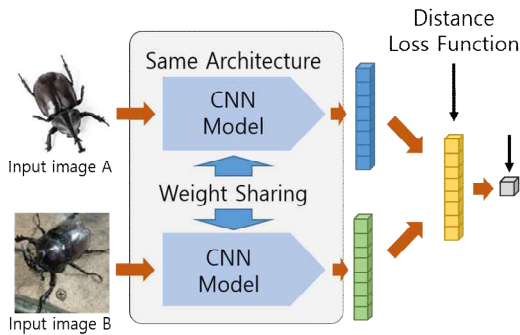


Fig. 1 Simple Siamese Networks Architecture.

이 자료를 CNN에 입력하여 온라인 학습에 사용한다 [14].

2.2 삼 네트워크(Siamese Network)기반 추적 알고리즘

삼 네트워크는 Fig. 1처럼 두 개의 동일한 구조로 구성된 신경망이다. 두 장의 입력 이미지를 동일한 네트워크로 입력한다. 동일한 커널을 사용하여 객체 특징을 추출하므로 가중치(weight)를 공유하는 특징을 지니고 있다. 추출된 특징은 L1 Norm, L2 Norm과 같은 거리 함수를 사용하여 유사도를 비교한다. 유사도 값에 따라 이미지의 차이를 평가하는 손실 값(Loss Value)을 얻을 수 있다. 이 손실 값을 줄여나가는 방식으로 삼 네트워크는 학습된다. 이 네트워크는 유사도 비교 분야인 얼굴 검증, 이미지 매칭에 적용됐다[15-16].

삼 네트워크를 추적 문제 해결에 적용하는 연구들이 진행되었다[17]. 이 연구에서는 추적 문제를 유사도 비교 문제로 간주하여 삼 네트워크를 적용하였다. 백본(Back-Bone) 네트워크로 AlexNet을 사용했다. 완전 연결 레이어를 제거하고 컨볼루션 레이어로 구성된 Fully CNN을 통해 입력 이미지에서 특징을 추출한다. 목표 객체 특징맵을 커널로 지정하고 찾기 영역에 교차 상관관계연산(cross-correlation operation)을 적용한다. 이 과정을 통해 유사도가 높은 객체 영역을 추정한다. GOTURN은 SiamFC와 동일하게 AlexNet을 삼 네트워크로 구성하여 객체 특징 추출에 사용했다[18]. 차이점은 완전 연결 레이어를 보존하였다. N-1번째 프레임과 N번째 프레임의 특징을 완전 연결 레이어 입력하여 목표 객체의 영역을 추정한다. 목표 객체가 장애물에 의해 가려지는 상황

에서 정확도를 높이기 위한 연구도 진행되었다. 연속 객체의 위치 추정을 위하여 삼 네트워크와 순환 신경망(Recurrent Neural Network)을 결합하였다[19].

3. 제안하는 객체 추적 모델

본 논문에서 제안하는 추적 알고리즘은 크게 3가지 모듈로 구성된다. 특징 추출에 사용되는 Fully CNN 기반 백본 네트워크, 삼 네트워크, 객체 영역 추출에 사용하는 지역제한 네트워크이다. 3가지 모듈이 유기적으로 결합되어 목표 객체를 추적한다.

컨볼루션 레이어는 FC 레이어 다음으로 연산량이 많이 요구된다. 본 논문에서 제안한 네트워크는 모든 레이어가 컨볼루션 레이어로 구성된다. 네트워크를 삼 네트워크 형태로 배치하므로 연산량은 단일 네트워크에 비해 2배 이상 요구된다. 연산량이 증가할수록 계산시간 또한 증가한다. 이러한 문제점을 해결하기 위해 네트워크를 보틀넥(Bottle Neck) 구조로 변환하여 설계했다. 보틀넥은 컨볼루션 레이어 구조 변환을 통해 파라미터 수를 축소하는데 효과적이다. 파라미터 개수는 이전 컨볼루션 레이어에서 출력된 특징맵 개수와 다음 컨볼루션 레이어의 커널 사이즈 및 커널 개수를 곱셈 연산을 통해 계산된다. 컨볼루션 레이어가 2개 이상일 경우에는 계산된 파라미터를 합산한다. 식 1은 컨볼루션 레이어에서 요구하는 파라미터 개수를 계산식을 나타낸다.

$$Parameters = \text{Number of Input Feature Maps} \times \text{Kernel Width} \times \text{Kernel Height} \times \text{Number of Out Feature Maps} \quad (1)$$

Fig. 2는 보틀넥 구조를 보여준다. 이 구조는 특징맵 압축, 특징추출, 특징맵 확장 단계를 포함한다. 특징맵 압축에는 1x1 크기의 커널로 이루어진 컨볼루션 연산을 사용한다. 이 커널은 입력 특징맵의 개수를 조절하는데 사용한다. 축소된 특징맵에 NxN 크기 커널을 적용하여 특징맵을 추출한다. 마지막 단계에서는 다시 1x1 크기의 커널을 적용하여 특징맵의 개수를 확장 시킨다. 보틀넥 구조를 적용하면 적층된 컨볼루션 레이어 사이의 파라미터 개수가 감소되므로 연산량 축소가 가능하다. Fig. 2의 상단은 보틀넥을 적용하기전 컨볼루션 연산이다. 필요 파라미터 수는 819,200개가 필요한 반면에 보틀넥을 적용할 경우 필요 파라미터 수가 221,184개로 약 3.7배 감소되는 결과를 보여준다. 파라미터 수가 줄어드는 것은 정보

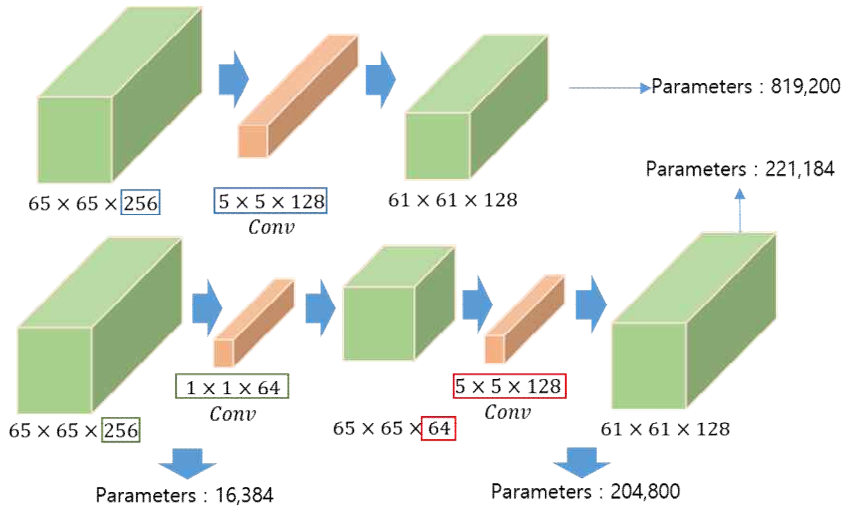


Fig. 2. Comparison of Number of Parameters Between Non-Bottleneck and Bottleneck.

손실이 발생할 수 있다. 정보 손실이 발생하는 것은 특징 정보가 사라지는 결과를 얻을 수 있다. 따라서 Fig. 3의 상단과 같이 입력 특징맵을 출력 특징맵에 연결(Concatenate)한다. 이 연산을 통해 특징 정보 흐름을 보존한다. 이 구조를 컨볼루션 블록으로 구성하여 적층한다.

Fig. 3 하단과 같이 컨볼루션 블록을 적층하여 백본 네트워크로 제작했다. 백본 네트워크를 Y자형 분기로 이루어진 삼 네트워크 형태로 구성했다. 삼 네트워크에는 두 장의 이미지가 입력된다. 목표 객체 이미지와 목표 객체가 포함된 찾기 영역 이미지다. 각 이미지는 네트워크를 통과하며 특징맵으로 출력된다. 마지막 레이어에서 출력된 특징맵을 사용하여 객체의 위치와 영역을 추정해야한다. 이를 위하여 지

역제안 네트워크(Region Proposal Network)를 서브 모듈로 연결 하였다. 지역 제안 네트워크는 객체 검출에 매우 효율적인 방법으로 알려져있다[20]. 이 네트워크는 이미지에 존재하는 특정 객체 및 영역을 추론한다. 객체 검출 방법과 달리 객체 추적 문제에서는 찾아야할 객체가 특정 학습 데이터셋에 존재하는 것이 아닌 초기 입력되는 첫 번째 프레임의 객체를 기준으로 검출하는 것이다. 지역 제안 네트워크의 핵심은 앵카(anchor) 박스다. 앵카박스를 적용함으로써 좌표 회귀(Regression)를 통해 박스 좌표를 추론하는 것과 박스 내부에 객체 존재 여부를 판별할 수 있다. 앵카 박스는 중점X(CenterX), 중점Y(CenterY), 넓이(Width), 높이(Height) 4가지 값을 포함한다. 본 논문에서는 5개의 앵카박스를 사용하여 객체 영역을 추정한다. Fig. 4는 백본네트워크와 지역

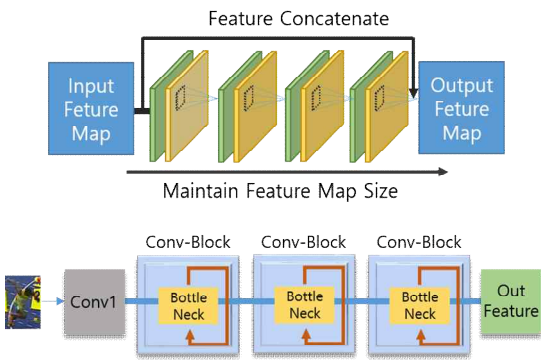


Fig. 3. Convolution Block with BottleNeck and Backbone Network.

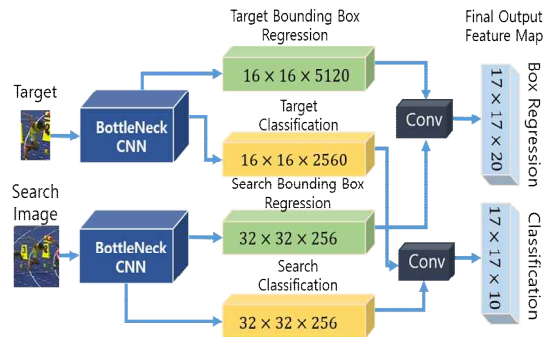


Fig. 4. Architecture of Region Proposal Network .

제안네트워크의 전체 구조도다. 네트워크의 최종 출력으로 17×17×20, 17×17×10 2개 특징맵을 출력한다. 각 특징맵은 박스 좌표 회귀 및 객체 존재 여부 값을 가지고 있다. 20은 5개 앙카박스와 4개의 좌표값이고, 10은 5개 앙카박스와 객체 존재 여부 값인 0, 1의 값이다.

4. 실험 결과

4.1 실험환경

본 논문에서는 제안한 객체 추적을 위한 네트워크의 성능을 검증했다. 실험에 사용된 하드웨어 및 소프트웨어 사양은 Table 1과 같다. 네트워크 및 알고리즘은 모두 Python 언어로 구성되어 있으며, 네트워크 설계, 학습, 추론 과정은 파이토치(Pytorch) 딥러닝 프레임워크를 사용했다.

네트워크 학습을 위해 ILSVRC(ImageNet Large Scale Visual Recognition Challenge) 2015 데이터셋을 사용하였다. 이 데이터셋은 3,862개의 학습 영상과 555개의 검증 비디오로 구성되어 있다. 각 비디오는 프레임 단위로 추출된 이미지로 구성되어 있다. 각 이미지는 주석파일로 포함하고 있다. 주석에 기입된 GT(Ground Truth)를 토대로 객체 영역을 추출한다. 네트워크는 목표 객체 이미지, 찾기 영역, 정규화된 GT 좌표, 객체 존재 유무 파라미터를 학습에 사용한다. 2장의 입력 이미지를 각 비디오별로 임의로 추출한다. 네트워크는 이미지 유사도를 학습하기 때문에 추출 순서는 무시된다. 목표 객체가 이미지의 정중앙에 위치되도록 배치하였다. 정규화된 GT 좌표는 식 (1)~(4)를 사용하여 앙카박스를 기준으로 재구성된다. 식에서 GT 값은 객체 좌상단 X, Y좌표값과 넓이(Width), 높이(Height)값이며, BOX 값은 앙카 박스 중점 X, Y좌표 값과, 박스의 넓이, 높이 값을 나타낸다. 각 GT 좌표는 전처리된 이미지의 새로운

Table 1. Specification of Experiment Environment.

		Version
H/W	O/S	Windows 10
	CPU	I7-8700K
	RAM	32G
	GPU	Titan XP
S/W	Deep learning Framework	Pytorch 1.8

객체 영역 좌표에 대응되도록 변환된다.

$$Regression_X = \frac{Ground Truth_X - BOX_X}{BOX_W} \quad (1)$$

$$Regression_Y = \frac{Ground Truth_Y - BOX_Y}{BOX_H} \quad (2)$$

$$Regression_W = LOG\left(\frac{Ground Truth_W}{BOX_W}\right) \quad (3)$$

$$Regression_H = LOG\left(\frac{Ground Truth_H}{BOX_H}\right) \quad (4)$$

앙카박스 내부의 객체 존재 유무는 Fig. 5처럼 1, 0, -1로 레이블을 지정한다. 레이블은 IoU(Intersection Over Union)을 사용하여 계산한다. IOU는 GT 영역과 앙카박스의 겹침 정도를 나타내는 척도이다. IoU가 50%이상일 때 객체가 존재하는 것으로서 해당 앙카 박스에 1을 지정한다. IoU가 40%이하일 경우 객체가 존재하지 않는 것으로 판단하고 0을 지정한다. IOU가 40%초과 50% 미만일 경우에 네트워크 학습에 혼란을 가중 할 수 있으므로 영향을 미치지 않도록 학습 과정에 해당 값을 무시하도록 -1을 할당한다.

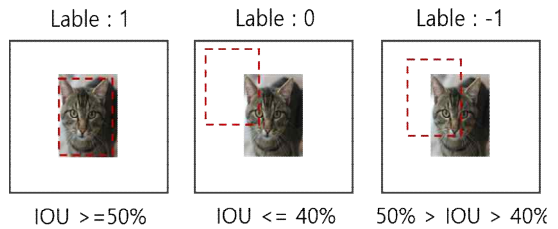


Fig. 5. Object Classification Label in Anchor Boxes.

4.2 성능 평가

알고리즘의 성능평가를 위해 50개 비디오로 구성된 OTB-2013[21] 데이터셋을 사용하였다. 해당 데이터셋에 포함된 영상 속성은 Table 2와 같다. 비디오는 최소 1개 이상의 속성을 포함한다. 평가 대상 알고리즘은 공개된 추적 알고리즘인 MKCFup[15], CSRDCF-LP[21]을 사용했다. 또한, 제안한 알고리즘과 동일한 환경에서 실험을 진행했다.

알고리즘의 정량적 평가를 위해 OPE(One Pass Evaluation)지표를 사용하였다. 이 지표는 Precision plot과 Success plot을 포함한다. Precision plot은 중점좌표에러를 계산한다. 추적 알고리즘을 통해 추정

Table 2. The Sequence Attribute Table for Algorithm Evaluation.

Attribute	Description
Illumination Variation(IV)	Change the lighting in the target object region
Scale Variation(SV)	Change in size of target object
Occlusion(OCC)	Object occlusion by obstacles
Deformation(DEF)	Target object deformation
Motion Blur(MB)	Blur on target object
Fast Motion(FM)	fast motion of target object
In-Plane Rotation(IPR)	Rotation of an target object within an image
Out-of-plane Rotation(OPR)	Rotation of an target object outsid an image
Out-of-View(OV)	Some areas of the target object move out of the image
Background Clutters(BC)	Generate target object-like color or texture
Low Resolution(LR)	Target object has low resolution

된 바운딩 박스의 중심 좌표와 GT(Ground Truth)의 바운딩박스 중심 좌표 거리 평균을 계산한다. Success plot는 추정된 바운딩 박스와 GT의 바운딩 박스 사이의 IoU이다.

Fig. 6은 OTB2013 데이터셋을 사용한 전체 성능 평가 결과를 보여준다. Fig. 6(a)는 중첩율 지표인 Success plot이고, Fig. 6(b)는 중첩에러율을 나타내는 Precision plot이다. Success plot 그래프 x축 값은 Overlap threshold로서 오버랩 비율을 나타낸다. 우측영역이 넓을수록 GT영역과 완벽하게 매칭 된다는 것을 의미한다. Precision plot 그래프 x축 값은 Location error threshold로 중점과 픽셀 간격 차이를

나타낸다. 좌측 0픽셀부터 우측 50 픽셀까지 값이 설정되어 있다. 해당 지표를 통해 추적 알고리즘이 목표 객체를 지속적으로 추적하는지 추적에 실패하는지를 평가 할 수 있다. 제안하는 알고리즘은 중첩율에서 0.611, 중점 에러율에서 0.831을 달성하였다.

Fig. 7은 Background Clutters속성의 Basketball 영상을 이용한 추적 결과이다. 적색 바운딩 박스가 제안하는 알고리즘이며 초록색은 GT 영역이다. 이 영상은 노이즈 및 움직이는 다중 객체들로 인하여 복잡한 배경을 포함하고 있다. 특히 649 프레임에서 카메라 플래시로 인하여 배경이 밝아지는 Illumination Variation현상까지 발생했음에도 불구하고 정

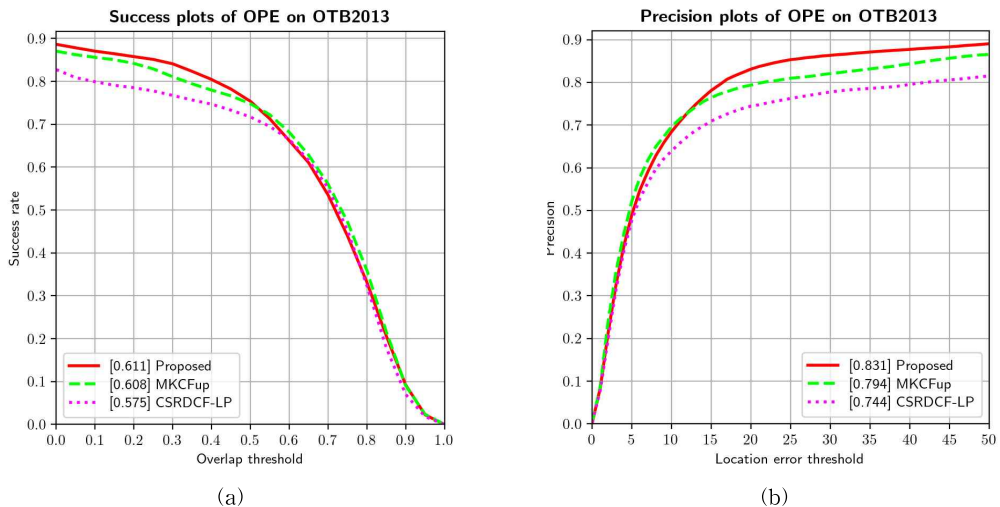


Fig. 6. Result of OTB2013 Dataset. (a) Result of Success Plot and (b) Result of Precision Plot.

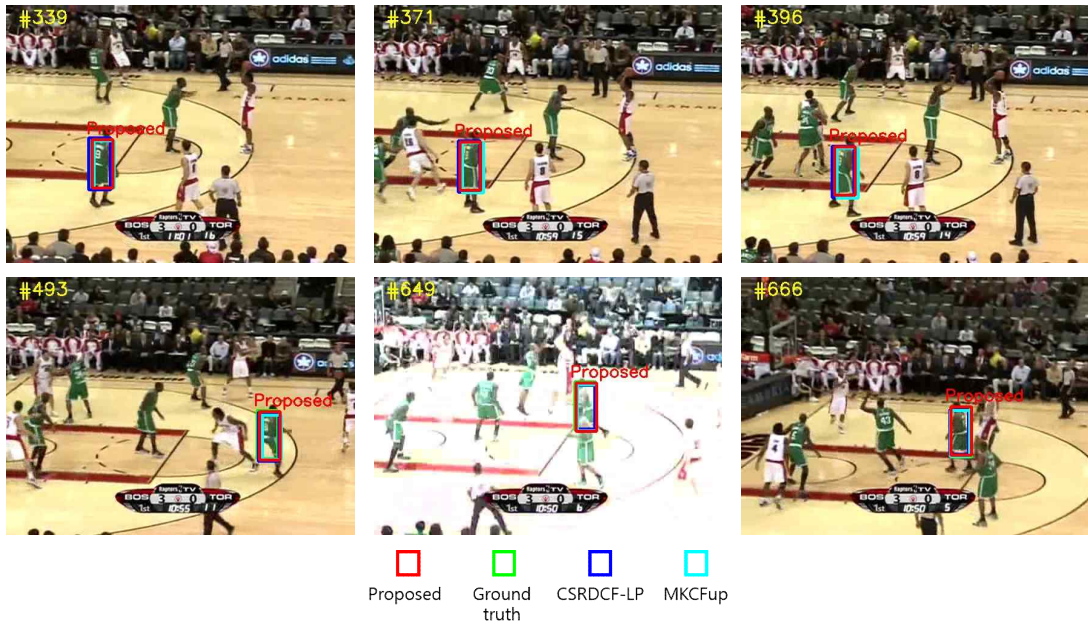


Fig. 7. Result of Background Clutters Attribute.

확하게 객체를 추적하는 결과를 볼 수 있다.

Fig. 8은 Low Resolution 속성을 지닌 CarScale 영상이다. 1 프레임부터 192 프레임까지 점진적으로 객체 크기 변화가 발생하고 있다. 156 프레임까지 제안하는 알고리즘이 목표 객체 영역 전체를 적색의 박스로 바운딩하고 있다. 하지만 192번 프레임의 초록색 GT 영역은 자동차의 트렁크 영역까지 포함되어 감싸고 있지만, 제안하는 알고리즘은 그 절반 크기만 추적하는 것을 관찰 할 수 있다. 초기화에 사용

된 1 프레임의 객체 영역을 보면 낮은 크기의 해상도로 구성되어 있어 샵 네트워크에서 추출되는 특징 정보에 제한이 발생한 것으로 사료된다. Table 3은 11가지 속성에 대한 전체 결과이다. 제안하는 알고리즘은 전체 성능으로 비교했을 때 가장 높은 값을 출력하였다. IPR, OCC, LR, MB, SV 속성에서 다른 알고리즘보다 낮은 수치를 보였지만 OV, IV, BC, FM, DEF, OPR 6개 속성에서 가장 높은 결과를 보여 주었다. Success Plot의 경우 11가지 속성중 OV 속

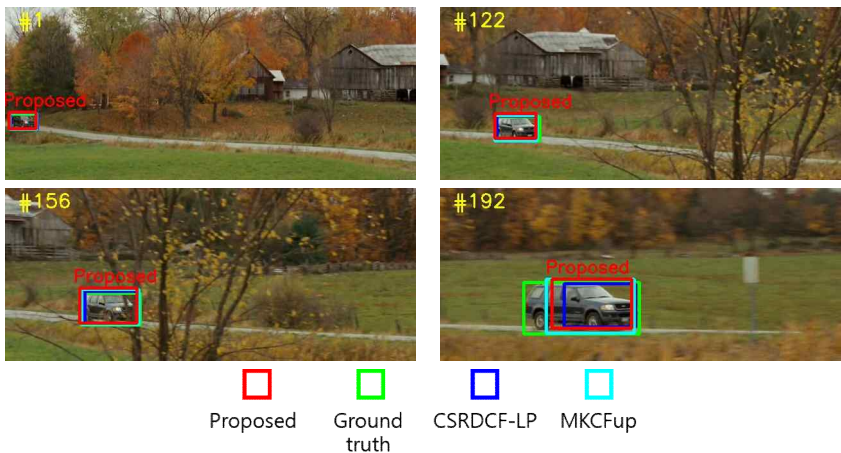


Fig. 8 Result of Low Resolution Attribute.

Table 3. Overall Performance Evaluation Results for Attributes of OTB2013 Benchmark Dataset.

		Proposed	MKCFup	CSRDCF-LP
Total	Success	[0.611]	0.608	0.575
	Precision	[0.831]	0.794	0.744
IPR	Success	0.542	[0.563]	0.524
	Precision	[0.742]	0.733	0.687
OCC	Success	0.631	[0.639]	0.559
	Precision	0.848	[0.856]	0.713
OV	Success	[0.635]	0.585	0.554
	Precision	[0.790]	0.702	0.683
IV	Success	[0.575]	0.552	0.529
	Precision	[0.780]	0.705	0.668
LR	Success	0.294	[0.487]	0.374
	Precision	0.542	[0.735]	0.568
BC	Success	[0.575]	0.518	0.512
	Precision	[0.783]	0.668	0.667
FM	Success	[0.564]	0.501	0.549
	Precision	[0.736]	0.633	0.691
MB	Success	0.548	[0.561]	0.547
	Precision	[0.733]	0.717	0.684
SV	Success	0.563	[0.570]	0.517
	Precision	[0.763]	0.729	0.653
DEF	Success	[0.630]	0.611	0.594
	Precision	[0.837]	0.805	0.783
OPR	Success	[0.598]	0.581	0.544
	Precision	[0.810]	0.767	0.716

성에서 0.635를 달성하며 가장 좋은 결과를 보여주었고, LR 속성에서 0.294를 달성하며 낮은 수치를 보여주었다.

5. 결 론

본 논문에서는 보틀넥을 적용한 CNN 기반 삼 네트워크 객체 추적 알고리즘을 제안하였다. 제안한 알고리즘에서는 객체 특징 강조와 위치 추정을 위한 지역 제안 네트워크 두 가지 구성 요소를 사용하여 객체 추적을 진행했다. 목표 객체 및 찾기 영역에서의 특징 강조는 보틀넥을 적용한 CNN을 사용했다. 보틀넥을 지나며 손실되는 정보를 보존하기 위하여 이전 레이어의 특징맵을 다음 레이어와 연결하여 정보 흐름을 보존하였다. 정보 흐름 보존 효과로 인하여 목표 객체의 특징이 강조되는 효과를 얻었으며,

이를 통해 객체 영역을 추정에 이점을 얻었다. 지역 제안 네트워크는 목표 객체와 찾기 영역의 특징맵을 사용하여 객체 후보 영역을 추정하고 객체 존재 여부를 판별한다. 객체가 존재하는 바운딩 박스의 영역을 합병하여 최종 객체 영역을 추정하였다. OTB2013 데이터셋을 사용한 실험 결과를 통해 Success Plot에서 0.611, Precision Plot에서 0.831을 달성했다. 반면 Low Resolution 속성에서 약점을 보여주었다. 목표 객체의 초기 해상도 사이즈가 작을 경우 객체 추적에 필요한 충분한 특징 정보를 얻지 못하여 추적에 실패한 것으로 사료된다.

향후 연구로는 저해상도 객체에서 의미론적인 정보를 출력하기 위한 특징 표현 알고리즘 연구와 네트워크 압축을 통해 임베디드 환경을 사용하는 플랫폼에서 실시간성을 보장받기 위하여 연산량 축소를 통한 연산속도 향상 알고리즘 연구를 진행하고자 한다.

REFERENCE

- [1] S. Kim, S. Lim, and D. Kim, "Intelligent Intrusion Detection System Featuring a Virtual Fence, Active Intruder Detection, Classification, Tracking, and Action Recognition," *Annals of Nuclear Energy*, Vol. 112, pp. 845–855, 2018.
- [2] V.A. Laurence, J.Y. Goh, and J.C. Gerdes, "Path-Tracking for Autonomous Vehicles at the Limit of Friction," *Proceeding of American Control Conference (ACC)*, pp. 5586–5591, 2017.
- [3] A. Koubâa and B. Qureshi, "DroneTrack: Cloud-Based Real-Time Object Tracking Using Unmanned Aerial Vehicles over the Internet," *IEEE Access*, Vol. 6, pp. 13810–13824, 2018.
- [4] S. Li and D.Y. Yeung, "Visual Object Tracking for Unmanned Aerial Vehicles: A Benchmark and New Motion Models," *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, Vol. 31, No. 1, pp. 4140–4146, 2017.
- [5] J. Li and J. Wang, "Adaptive Object Tracking Algorithm based on Eigenbasis Space And Compressive Sampling," *IET Image Processing*, Vol. 6, No. 8, pp. 1170–1180, 2012.
- [6] S.D. Lin, J.J. Lin, and C.Y. Chuang, "Particle Filter With Occlusion Handling for Visual Tracking," *IET Image Processing*, Vol. 9, No. 11, pp. 959–968, 2015.
- [7] S.K. Weng, C.M. Kuo, and S.K. Tu, "Video Object Tracking Using Adaptive Kalman Filter," *Journal of Visual Communication and Image Representation*. Vol. 17, No. 6, pp. 1190–1208, 2006.
- [8] S. Lim and D. Kim, "Object Tracking Using Feature Map from Convolutional Neural Network," *Journal of Korea Multimedia Society*, Vol. 20, No. 2, pp. 126–133, 2017.
- [9] N. Wang and D.Y. Yeung, "Learning A Deep Compact Image Representation for Visual Tracking," *Advances in Neural Information Processing Systems*, 2013.
- [10] S. Park, S. Kim, S. Lim, and D. Kim, "Performance Comparison of Commercial and Customized CNN for Detection in Nodular Lung Cancer," *Journal of Korea Multimedia Society*, Vol. 23, No. 6, pp. 729–737, 2020.
- [11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision And Pattern Recognition*, pp. 770–778, 2016.
- [12] S. Hong, T. You, S. Kwak, and B. Han, "Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network," *Proceedings of the International Conference on Machine Learning*, pp. 597–606, 2015.
- [13] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.H. Yang, "Hedged Deep Tracking," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4303–4311, 2016.
- [14] K. Zhang, Q. Liu, Y. Wu, M.H. Yang, "Robust Visual Tracking via Convolutional Networks without Training," *IEEE Transaction on Image Process*, Vol. 25, No. 4, pp. 1779–1792, 2016.
- [15] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. "Deepface: Closing The Gap to Human-Level Performance in Face Verification," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, 2014.
- [16] T.Y. Lin, Y. Cui, S. Belongie, J. Hays, and C. Tech, "Learning Deep Representations for Ground-to-Aerial Geolocation," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5007–5015, 2015.
- [17] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H. Torr, "Fully Convolutional Siamese Networks for Object Tracking," *Proceedings of the European Conference on*

Computer Vision, pp. 850-865, 2016.

[18] D. Held, S. Thrun, and S. Savarese. "Learning to Track at 100 fps with Deep Regression Networks," *European Conference on Computer Vision*, pp. 749-765, 2016.

[19] D. Gordon, A. Farhadi, and D. Fox, "Re3: Real-Time Recurrent Regression Networks for Visual Tracking of Generic Objects," *IEEE Robotics and Automation Letters*, Vol. 3, No. 2, pp. 788-795, 2018.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, Vol. 28, pp. 91-99, 2015.

[21] W. Yi, J.W. Lim, and M.H. Yang, "Online Object Tracking: A Benchmark," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411-2418, 2013.

[22] M. Tang, B. Yu, F. Zhang, and J. Wang, "High-speed Tracking with Multi-kernel Correlation Filters," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4874-4883, 2018.

[23] A. Lukezic, T. Vojir, L. Cehovin, J. Matas, and M. Kristan, "Discriminative Correlation Filter with Channel and Spatial Reliability," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6309-6318, 2017.



임 수 창

2015년 순천대학교 컴퓨터공학과 졸업(공학사)
 2017년 순천대학교 대학원 컴퓨터공학과 졸업(공학석사)
 2019년 순천대학교 대학원 컴퓨터공학과 박사수료

2020년~현재 (주)티이에프 기업부설연구소 책임연구원
 관심분야: 컴퓨터비전, 영상처리, 딥러닝, 객체추적, 시계열분석



김 중 찬

2000년 순천대학교 컴퓨터공학과 졸업(이학사)
 2002년 순천대학교 대학원 컴퓨터공학과 졸업(이학석사)
 2007년 순천대학교 대학원 컴퓨터공학과 졸업(이학박사)

2013년 서울대학교 자동화 시스템 연구소 선임연구원
 2021년~현재 순천대학교 컴퓨터공학과 조교수
 관심분야: 영상처리, HCI, 콘텐츠, 컴퓨터그래픽스, 기계학습, 객체추적, 컴퓨터비전, 데이터 분석 및 예측